

#4

PATENTS

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants: Xiao Wei Hu, et al.

Serial No.: unassigned

Art Unit: unassigned

Filed: herewith

Docket: 14154

For: A FRAMEWORK SYSTEM AND
METHOD FOR TESTING
SERVER PERFORMANCE

Dated: January 11, 2001



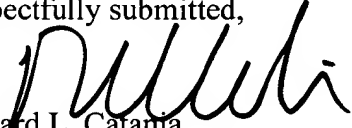
Assistant Commissioner for Patents
Washington, DC 20231

CLAIM OF PRIORITY

Sir:

Applicants, in the above-identified application, hereby claim the right of priority in connection with Title 35 U.S.C. §119 and in support thereof, herewith submits a certified copy of Chinese Patent Application No. 00 1 00949.4, filed on January 11, 2000.

Respectfully submitted,


Richard L. Catania
Registration No. 32,608

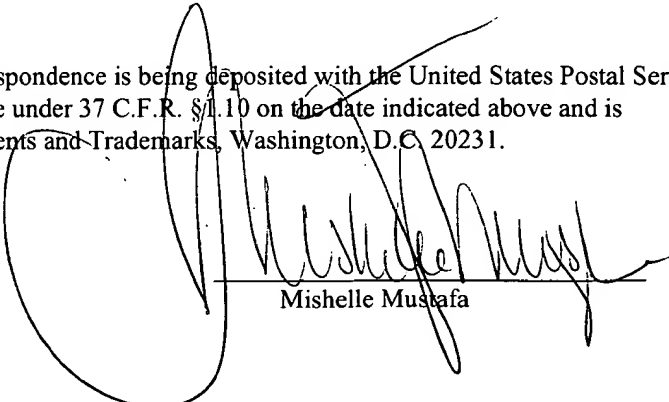
Scully, Scott, Murphy & Presser
400 Garden City Plaza
Garden City, NY 11530
(516) 742-4343
RLC:vjs

CERTIFICATE OF MAILING BY "EXPRESS MAIL"

"Express Mail" Mailing Label Number: EL754203663US
Date of Deposit: December 11, 2001

I hereby certify that this correspondence is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 C.F.R. §1.110 on the date indicated above and is addressed to the Assistant Commissioner of Patents and Trademarks, Washington, D.C. 20231.

Dated: December 11, 2001


Mishelle Mustafa

CERTIFIED COPY OF
PRIORITY DOCUMENT

JP 9- 1999- 0264

US (YOR)

证 明

1c914 U.S. PTO
09/759837
01/11/01

本证明之附件是向本局提交的下列专利申请副本

申 请 日: 2000 01 11

申 请 号: 00 1 00949.4

申 请 类 别: 发明专利

发明创造名称: 一种测试服务器性能的框架系统和方法

申 请 人: 国际商业机器公司

发明人或设计人: 扈晓炜; 冯南; 陈晓艳

中华人民共和国
国家知识产权局局长

姜 颖

2000 年 5 月 29 日

权 利 要 求 书

1. 一种测试具有混合工作负载的服务器的方法，其中作为代理机及控制设备的多个客户机通过网络与被测试的服务器相连，其特征在于：包括如下步骤，

第三方开发与其一种或多种工作负载类型对应的一个或多个工作负载类型配置模块，这些模块均实现一个工作负载类型配置模块接口；

第三方开发与其一种或多种工作负载类型对应的一个或多个工作负载引擎，这些引擎均实现一个工作负载接口；

控制设备通过所述工作负载类型配置模块接口使用所述工作负载类型配置模块进行工作负载配置，将配置过程中收集的信息发送到相应的代理机；

所述代理机通过所述工作负载接口控制相应的工作负载引擎，使所述工作负载引擎根据配置过程中收集的信息生成工作负载请求，并将请求发送到服务器；

所述控制设备从各代理机收集服务器的响应信息，并生成测试结果。

2. 根据权利要求 1 所述的测试方法，其特征在于：所述工作负载类型配置模块接口包括使框架系统调用第三方工作负载类型配置模块为不同的测试目的定义工作负载类型的功能。

3. 根据权利要求 1 所述的测试方法，其特征在于：所述工作负载接口包括进行工作负载建立，控制及监视的功能。

4. 根据权利要求 1-3 任一项所述的测试方法，其特征在于：所述一个或多个工作负载类型配置模块位于控制设备上，所述一个或多个工作负载引擎被分别设置到一个或多个代理机上。

5. 根据权利要求 4 所述的测试方法，其特征在于：由位于所述控制设备上的一个控制器来实现该控制设备所进行的配置过程，并与所

述代理机之间进行信息传送。

6. 根据权利要求 5 所述的测试方法，其特征在于：由位于所述代理机上的一个适配器对工作负载引擎进行控制并与控制设备通信。

7. 根据权利要求 5 所述的测试方法，其特征在于：所述控制器进行的配置的过程包括如下步骤，

从一个可用的工作负载列表中选择一种工作负载类型；

根据所选择的工作负载类型激活对应的工作负载类型配置模块；

所述一个或多个工作负载类型配置模块对各工作负载类型进行配置，并将配置信息传输给控制器；

所述控制器分别指定将要生成所述工作负载类型下的工作负载请求的一个或多个代理机的地址和每个代理机所模拟的客户数。

8. 根据权利要求 7 所述的测试方法，其特征在于：所述工作负载类型配置模块为所选择的工作负载类型收集产生工作负载请求时所需的相关信息。

9. 根据权利要求 1-3 或 8 所述的测试方法，其特征在于：所述配置过程中收集的信息至少包括工作负载类型及配置信息，和代理机所模拟的客户数。

10. 根据权利要求 6 所述的测试方法，其特征在于：代理机适配器通过所述工作负载接口将接收的信息传输给工作负载引擎；工作负载引擎将接收的服务器动态响应信息发送回代理机适配器；

11. 根据权利要求 1-3 及 10 任一项所述的测试方法，其特征在于：进一步包括如下步骤，所述控制器通过向代理机发送命令控制测试工作的开始和结束。

12. 一种测试具有混合工作负载的服务器的测试框架系统，其中作为代理机及控制设备的多个客户机通过网络与被测试的服务器相连，其特征在于：在控制设备一侧具有

控制器，用于协调参加测试的各部件；

工作负载类型配置模块接口，使第三方开发的一个或多个工作负

载类型配置模块能够加入到该框架系统中;

所述工作负载类型配置模块,使第三方描述自己具体的测试需求,

在代理机一侧具有

代理适配器,用于从控制设备接收命令和配置信息,并把服务器的响应信息返回给控制器;

工作负载引擎接口,使第三方开发的一个或多个工作负载引擎能够加入到该框架系统中;

所述工作负载引擎,用于接收来自于代理适配器的命令和配置信息,生成工作负载请求,将该请求向服务器发送,并从服务器接收响应信息。

13. 根据权利要求 12 所述的测试框架系统,其特征在于:所述工作负载类型配置模块接口包括使框架系统调用第三方工作负载类型配置模块为不同的测试目的定义工作负载类型的功能。

14. 根据权利要求 12 所述的测试框架系统,其特征在于:所述工作负载接口包括进行工作负载建立,控制及监视的功能。

15. 根据权利要求 12-14 任一项所述的测试框架系统,其特征在于:所述控制器通过所述工作负载类型配置模块接口使用一个或多个工作负载类型配置模块进行工作负载类型配置,指定将要生成所述工作负载类型下的工作负载请求的一个或多个代理机的地址和每个代理机所模拟的客户数,将在配置过程中所收集的信息通过控制器传输给对应于相应的一个或多个代理机地址的代理机适配器。

16. 根据权利要求 12-14 任一项所述的测试框架系统,其特征在于:所述工作负载类型配置模块为相应的工作负载类型收集生成相应的工作负载请求时所需的相关信息。

17. 根据权利要求 12-14 任一项所述的测试框架系统,其特征在于:所述发送给代理机的信息至少包括工作负载类型及配置信息,和代理机所模拟的客户数。

18. 根据权利要求 12-14 任一项所述的测试框架系统,其特征在于:

所述控制器通过向代理机发送命令控制测试工作的开始和结束。

19. 根据权利要求 12-14 任一项所述的测试框架系统,其特征在
于:所述控制器动态地从各代理机接收服务器的响应信息。

20. 根据权利要求 12-14 任一项所述的测试框架系统,其特征在
于:所述控制设备还具有,

一个工作负载类型库,用于存储第三方所定义的工作负载类型
及控制器在配置过程中所收集的信息。

21. 根据权利要求 20 所述的测试框架系统,其特征在在于:所述控
制设备还具有,

一个工作引擎库,用于存储第三方所定义的工作引擎。

22. 根据权利要求 12-14 任一项所述的测试框架系统,其特征在
于:每个所述代理机上至少有一个工作负载引擎。

说明书

一种测试服务器性能的框架系统及方法

技术领域

本发明涉及一种用于测试服务器性能的框架系统及应用该框架系统进行服务器性能测试的方法，尤其涉及一种对具有混合工作负载的服务器性能进行测试的框架系统及方法。

背景技术

随着服务器能力的日益强大，可以将不同类型的多个工作负载放在一台服务器上运行。然而，由于计算机资源，例如CPU、存储器容量等的限制，这些不同类型的工作负载可能会相互影响，对于一个企业来讲，为了达到商业目的，重要的是要使服务器具有平衡工作负载的能力，而在实现这一目的之前需要评价和测试服务器的能力。

现有技术中一种典型的基准测试系统如图1所示，该测试系统包括一组用于产生工作负载请求的生成器，一个协调测试工具的测试工作的管理器及作为测试对象的服务器，在这种测试环境下，仅能测试单一的工作负载，或者仅能测试一个具体的应用系统的一个具体的工作负载，而不能测试混合工作负载。

美国专利US 5, 790, 425公开了一种在客户机服务器环境下对服务器进行测试的方法，该方法可以在具有不同协议的客户系统中实现，并且可以测试服务器对多种工作负载请求的响应。但是该对比文件未教导如何将第三方的测试要求加入其测试方法中，以根据第三方的测试目的进行测试。

发明目的

发明目的在于提供一种对具有混合工作负载的服务器性能进行测试的框架系统及方法，通过采用本发明的框架进行测试，可以将第三方的测试要求灵活地加进来，并根据第三方的测试要求测试服务器

的能力。

技术方案

为实现上述目的，本发明提供一种测试具有混合工作负载的服务器的方法，其中作为代理机及控制设备的多个客户机通过网络与被测试的服务器相连，包括如下步骤，第三方开发与其一种或多种工作负载类型对应的一个或多个工作负载类型配置模块，以实现一个工作负载类型配置模块接口；第三方开发与其一种或多种工作负载类型对应的一个或多个工作负载引擎，以实现一个工作负载接口；控制设备通过所述工作负载类型配置模块接口使用所述工作负载类型配置模块进行配置，将配置过程中收集的信息发送到相应的代理机；所述代理机通过所述工作负载接口控制相应的工作负载引擎，使所述工作负载引擎根据配置过程中收集的信息生成工作负载请求，并将请求发送到服务器；所述控制设备从各代理机收集服务器的响应信息，并生成测试结果。

本发明还提供一种测试具有混合工作负载的服务器的测试框架系统，其中作为代理机及控制设备的多个客户机通过网络与被测试的服务器相连，其特征在于：在控制设备一侧具有控制器，用于协调参加测试的各部件；工作负载类型配置模块接口，使第三方开发的一个或多个工作负载类型配置模块能够加入到该框架系统中；所述工作负载类型配置应用，使第三方描述自己具体的测试需求，在代理机一侧具有代理适配器，用于从控制设备接收命令和所述配置信息，并把测试结果返回给控制器；工作负载引擎接口，使第三方开发的一个或多个工作负载引擎能够加入到该框架系统中；所述工作负载引擎，用于接收来自于代理适配器的命令和配置信息，生成工作负载请求，将该请求向服务器发送，并从服务器接收响应信息。

技术效果

本发明的框架系统可以灵活地将第三方新的工作负载加入测试框架之中，而不管这种工作负载是一种标准的基准测试工作负载或是专门的应用工作负载，从而可以对具有混合工作负载的服务器性能进

行测试。

通过以下结合附图对本发明所进行的详细描述，本发明的优点将更加清楚。

图1是现有技术中的测试系统。

图2是根据本发明的测试框架系统的概略方框图。

图3是采用本发明的测试框架系统进行服务器性能测试的方法流程图。

图4是本发明的一个实施例的测试框架系统的方框图。

本发明的详细描述

本发明用于测试服务器的框架系统如图2所示。

其中控制设备100是用于协调框架系统中其它部件操作的设备。控制器101用于帮助第三方将自己的测试需求加进框架系统，并负责与后述的代理机进行通信。工作负载类型配置模块接口102定义了一组控制器101可以调用的功能，这些功能由要使用本发明的框架系统进行测试的第三方来实现，这样第三方就可以将自己的测试工作负载类型灵活地加入框架系统。在本发明的一个实施例中，该接口是用Java语言实现的。对于本领域的技术人员来讲，也可以用其它任何高级语言实现该接口功能。工作负载类型配置模块103是第三方自己编制的，用于配置具体的测试需求，并且该工作负载类型配置模块103通过实现工作负载配置模块接口102中所定义的所有功能，来实现与控制器101之间的通信。工作负载类型库104用于存储第三方所定义的所有工作负载类型，及工作负载类型配置模块103在进行工作负载配置时所收集的信息。工作负载类型配置模块103可以对该工作负载类型库104中可用的工作负载类型下的工作负载进行配置，并将配置工作负载类型时所收集的信息发送给控制器101。

代理机200，代理机300...代理机400由分布于网络中的客户机实现，用于模拟测试过程中向待测服务器500发出工作负载请求的用户。代理适配器201，301...401分布在各代理机上，用于从控制器接收命令并控制后述的工作负载引擎203，303...403生成测试服务器的

工作负载请求。工作负载接口202、302...402定义了一组代理适配器201, 301...401可以调用的功能, 这些功能由要使用本发明的框架系统进行测试的第三方来实施, 这些功能定义了如何设置、控制后述的工作负载引擎并报告测试结果等功能。在本发明的一个实施例中, 该接口是用Java语言来实现的。但是对于本领域的技术人员来讲, 当然也可以用其它任何与平台无关的高级语言来实现该接口功能。工作负载引擎203, 303...403是第三方自己编制的, 它能够根据第三方所要测试的工作负载类型及与该工作负载类型下的工作负载具体配置生成工作负载请求。并且该工作负载引擎203, 303...403同样也应实现工作负载接口202, 302...402中所定义的所有功能, 从而使工作负载引擎203, 303...403与代理适配器201, 301...401进行通信。

以上描述了根据本发明所实现的测试框架系统, 下面将根据图3描述使用根据本发明的框架系统测试服务器对混合工作负载请求响应的能力的服务器的方法。

首先在步骤S1, 测试者(第三方)要在控制设备100和代理机200, 300...400及服务器上设置适当的软件, 这些软件应当实现如下功能: 即控制器101, 代理适配器201、301...401的功能。另外, 测试者也应将欲参加测试的工作负载的类型信息及在产生工作负载请求时可能要用到的相关信息分别存储到相应的控制设备的工作负载类型库104, 代理机的存储器(未示出)及服务器500上。

在步骤S2, 测试者要根据自己的测试需求编制与参加测试的各工作负载类型相对应的工作负载类型配置模块103, 这些工作负载类型配置模块103应当实现工作负载类型配置模块接口102中定义的功能以与控制器101进行通信。

在步骤S3, 测试者编制与参加测试的各工作负载类型相对应的工作负载引擎203, 303...403, 该工作负载引擎通过使用工作负载类型及该类型下工作负载配置信息生成请求。同样, 这些工作负载引擎203, 303...403应当实现工作负载接口202, 302...402中定义的功能以与代理适配器201、301...401进行通信。另外, 在图2中为了说明的方

便将工作负载接口用202, 302, 402标记, 但实际上它们定义了相同的一组功能。

在步骤S4, 控制器101通过工作负载类型配置模块接口102使用一个相应的工作负载类型配置模块103对工作负载进行配置, 并收集配置过程中收集的信息。测试者可根据自己的需求重复执行步骤S4对多种工作负载类型进行配置。

在步骤S5, 控制器101将配置过程中收集的信息分别发送到相应的代理机适配器201, 301...401上。

在步骤S6, 各代理机适配器201, 301...401分别通过工作负载接口202, 302...402控制各工作负载引擎203, 303...403生成工作负载请求, 并将请求发送到服务器500。

在步骤S7, 控制器101从代理机201, 301...401动态地采集服务器500的响应信息并生成测试结果。

另外, 虽然在上面是以一个代理机上有一个工作负载引擎来描述的, 但是对于本领域技术人员来说, 当然也可以在一个代理机上设置多个工作负载引擎。

从以上的描述可知, 本发明的框架系统实现了灵活地测试具有混合工作负载下的服务器的功能。而这种灵活性主要体现在: 本框架系统使用工作负载类型配置模块接口和工作负载接口, 从而使得第三方可以随时将自己的工作负载加入测试任务中。

下面将根据一个具体的实施例来描述第三方如何采用本发明的测试框架系统进行测试。

现在假设要测试一个电子商务网站对混合工作负载请求的响应能力。

一般来讲, 一个电子商务站点(服务器)可能要承载许多工作负载, 有些工作负载是由偶然的浏览所产生的, 而另一些工作负载则是由进行商业交易的用户产生的, 并且在实际情况中, 前者出现的机会多于后者, 即仅仅到电子商务站点上进行浏览的用户更多一些。如果站点对这种情况没有适当的控制, 则偶然的浏览可能会降低进行商

业交易这种工作负载的性能。因此电子商务站点的测试者（第三方）要模拟实际的环境测试服务器和应用以了解在不同的因特网接入方式下服务器是如何响应的，从而有目的地在以后进行控制。

在应用本发明的框架系统进行测试时，首先，测试者在测试前要做如下工作：

1. 搭建起进行测试所需的硬件环境。如图4所示，这里假设测试者提供了一台可以运行混合工作负载的服务器6000，五台PC机即PC1-PC5，其中一台PC机即PC5作为控制设备1000而另外四台PC机即PC1-PC4作为代理机，它们之间通过网络连接。

2. 在控制设备1000（PC5）上加载实现控制器1001功能的软件，在代理机2000-5000（PC1-PC4）上加载实现代理适配器2001-5001功能的软件。

3. 测试者确定测试目的，在本实施例中，假设要测试同时运行了两种不同类型的工作负载的服务器，一种工作负载类型是HTTP，另一种工作负载类型是websurf。并且假设PC1和PC2用于模拟产生HTTP类型的工作负载请求，PC2, PC3和PC4用于模拟产生websurf类型的工作负载请求。

4. 测试者要将工作负载类型为HTTP下的工作负载的相关信息（例如要产生这种类型下的工作负载所需的变量，及变量的实例值）存放于控制设备1000的工作负载类型库1005中。

5. 按照与上述4相同的方法将工作负载类型为websurf下的工作负载的相关信息也存放于控制设备1000的工作负载类型库1005中。

6. 测试者开发与工作负载类型HTTP所对应的HTTP工作负载类型配置模块1003，该模块能够实现本发明的测试框架系统的工作负载类型配置模块接口1002，从而控制器1001可以通过该工作负载类型配置接口1002调用该HTTP工作负载类型配置模块1003进行工作负载配置。测试者可以根据HTTP类型及相关信息开发生成相应的工作负载请求的HTTP工作负载引擎2003、3003（这两个工作引擎是相同的），该工作负载引擎2003、3003能够实现本发明测试框架系统中的工作负载

接口2002、3002，从而控制器1001可以通过代理适配器2001、3001控制HTTP工作负载引擎2003、3003生成HTTP工作负载请求。测试者然后将HTTP工作负载类型配置模块1003加载到控制设备1000上，将HTTP工作负载引擎2003、3003加载到PC1和PC2上。这样测试者通过实现工作负载类型配置模块接口1002及工作负载接口2002、3002就实现了将自己的HTTP工作负载加入本发明的测试框架系统中。

7. 按照与6相同的操作，测试者开发与工作负载类型websurf所对应的websurf工作负载类型配置模块1004，该模块能够实现本发明的测试框架系统的工作负载类型配置模块接口1002，从而控制器1001可以通过该工作负载类型配置接口1002调用该websurf工作负载类型配置模块1004进行工作负载配置。测试者可以根据websurf类型及相关信息开发生成相应的工作负载请求的websurf工作负载引擎3004、4003、5003(这三个工作引擎是相同的)，该工作负载引擎3004、4003、5003能够实现本发明测试框架系统中的工作负载接口3002、4002和5002，从而控制器1001可以通过代理适配器3001、4001和5001控制websurf工作负载引擎3004、4003、5003生成websurf工作负载请求。测试者然后将websurf工作负载类型配置模块1004加载到控制设备1000上，将websurf工作负载引擎3004、4003、5003加载到PC2、PC3和PC4上。这样测试者通过实现工作负载类型配置模块接口1002及工作负载接口3002、4002HE 5002将与websurf工作负载类型相对应的websurf工作负载类型配置模块1004和websurf工作负载引擎3004、4003、5003加入本发明的测试框架系统中。

另外，为了更清楚地理解本发明的两个接口，现给出利用Java语言对本发明的框架系统的接口进行的示例定义。

工作负载类型配置接口的定义：

```
public interface iConfig {
    // *控制器将一个工作负载类型配置文件名传送给工作负载类型配置模块*//
    public void setConfigFileName (String strConfigFileName);
```

//* 配置工作负载类型*//

public Boolean show (frame parent) throws Exception;

// *收集配置信息*//

public string [] getAllFilesNeedToBeSend();

}

工作负载引擎接口的定义:

public interface IWorkload {

//*可以根据参数“configFileName”所表示的配置文件增加准备测试的一些初始化代码*//

public void initWorkload (String configFileName) throws Exception;

//*开始测试*//

public void start () throws Exception;

//* 结束测试*//

public void stop () throws Exception;

//*设置代理机上所模拟的用户数，及产生工作负载请求的间隔.*//

public void setproperty (Int nClientNumber, int nTime) throws Exception;

//*改变所模拟的用户数*//

public void changeClientNumber (int newClientNumber) throws Exception;

//*动态地报告对服务器的响应的统计信息，这些信息可以包括但不限于如下信息即，测试时间内总的吞吐量、总响应时间、总的连接量，一种错误类型下总的错误数，从测试开始时刻到当前时间的测试期间等*//

public CReturnValue getStandardValue() throws Exception;

//*结束测试*//

public boolean testFinish() throws Exception;

}

对于本领域的技术人员来讲，本发明的测试框架系统对于接口的定义是不限于上述例子限定的，接口中可以加入新的功能。

至此，根据本发明测试具有混合工作负载的框架系统已经构建完毕，下面就可以使用该框架系统进行测试。

本实施例中搭建好的框架系统如图4所示。关于图4中各块之间的关系及各块所完成的功能将在下面描述。

首先，测试者启动控制器1001以增加新的待测的工作负载类型，控制器1001从一个可用的工作负载列表（未示出）中选择一种工作负载类型例如HTTP，其中该工作负载列表可以由控制器1001根据工作负载类型库1004中所存储的可用的工作负载类型生成，也可以由测试者制作，关于该列表的实现，属于本领域技术人员所公知的内容，因此在此省略其说明。

然后控制器1001通过工作负载类型配置模块接口1002调用与HTTP工作负载类型对应的HTTP工作负载类型配置模块1003，该HTTP工作负载类型配置模块1003对工作负载类型HTTP下的工作负载进行配置，例如收集工作负载引擎2003、3003生成工作负载请求时所需的相关信息，并将其传输给控制器1001。另外工作负载类型配置模块1003在配置过程中所收集的相关信息也可以存放在一个HTTP工作负载类型文件中，并将该文件存放到工作负载类型库1005内，以备控制器1001调用。另外，该配置好的工作负载类型HTTP也被添加到一个工作负载类型列表中（未示出），该工作负载类型列表用于存放测试框架中可用的工作负载类型。

在配置完HTTP工作负载类型的工作负载后，控制器1001按照与配置HTTP类型时相似的过程，来配置websurf工作负载类型下的工作负载，并把配置信息传送给控制器，或/并且把配置信息存入到一个websurf工作负载类型文件中，以备控制器1001调用。另外，该工作负载类型websurf也被添加到上述工作负载类型列表中。

接着，控制器1001分别为添加到框架中的各工作负载类型指定生成该类型下工作负载请求的代理机。在本实施例中，即是为HTTP类型下的工作负载配置代理机PC1及PC2的网络地址，为websurf类型下的工作负载配置代理机PC2，PC3及PC4的网络地址，并/或将配置的

网络地址添加到一个机器列表（未示出）中，该机器列表用于存放可以生成某种类型下的工作负载请求的代理机网络地址；同时，控制器1001还能各代理机指定所模拟的用户数目，这样一台代理机可以模拟多个用户发出多个同一类型的工作负载请求。

接着，控制器1001通过向PC1、PC2、PC3及PC4发出测试开始指令，将HTTP工作负载类型下的工作负载配置信息及PC1 2000所模拟的用户数目通过网络发送到PC1的代理适配器2001；将HTTP工作负载类型下的工作负载配置信息及PC2所模拟的用户数目通过网络发送到PC2的代理适配器3001；将websurf工作负载类型下的工作负载配置信息及PC2所模拟的用户数目通过网络发送到PC2的代理适配器3001；将websurf工作负载类型下的工作负载配置信息及PC3所模拟的用户数目通过网络发送到PC3的代理适配器4001；将websurf工作负载类型下的工作负载配置信息及PC4所模拟的用户数目通过网络发送到PC4的代理适配器5001。

随后，PC1上的代理适配器2001将所接收到的工作负载配置信息及PC1所模拟的用户数目通过HTTP工作负载接口2002发送给HTTP工作负载引擎2003，该引擎于是产生HTTP工作负载请求，并将请求发送到服务器6000，并且从服务器接收响应信息进行统计，代理适配器2001然后将HTTP工作负载引擎2003所统计的测试信息发送回控制器1001。按照同样的操作，最终控制器1001收集到HTTP工作负载引擎3003，websurf工作负载引擎3004，websurf工作负载引擎4003及websurf工作负载引擎5003所统计的测试信息。

最后，控制器1001根据所有的测试信息生成测试结果，并将结果输出，在本实施例中，测试结果是实时地输出的。

另外，控制器1001还可以通过向各代理机发送停止命令来结束测试。

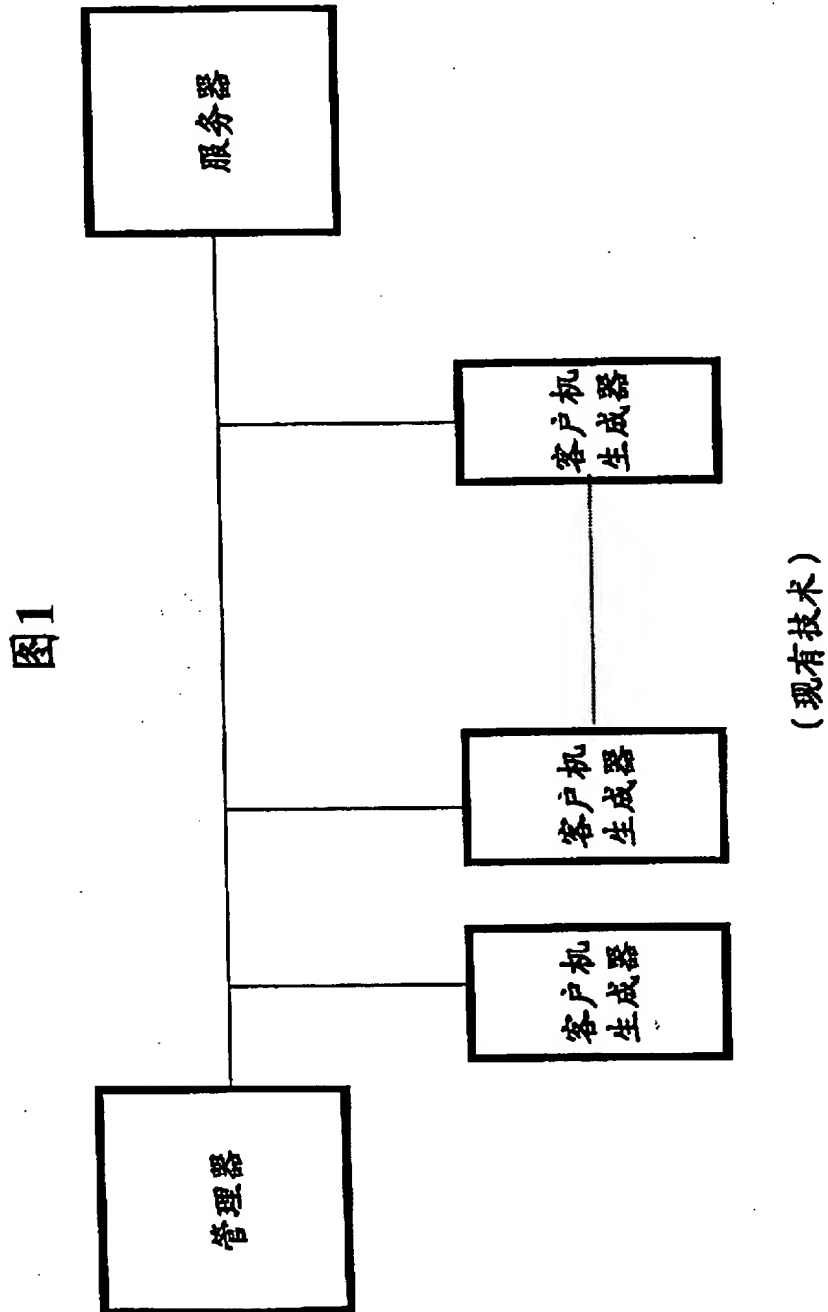
在控制器1001将配置信息发送到各代理机之前，还可以把定义好的测试环境信息包括可用的工作负载类型，可用的机器列表等存储到一个文件中，当需要再次在该测试环境中进行测试时，直接从该文

件中读取环境信息即可，而不需重新定义。

此外，在控制设备中还可以具有一个工作负载引擎库，用于存储第三方所定义的工作负载引擎，以备控制器使用。

本发明已经根据具体的实施例和附图进行了详细描述，但这些描述并非用于限制本发明。在不偏离本发明的精神和范围内，可以对其进行修改和改进。

说明书附图



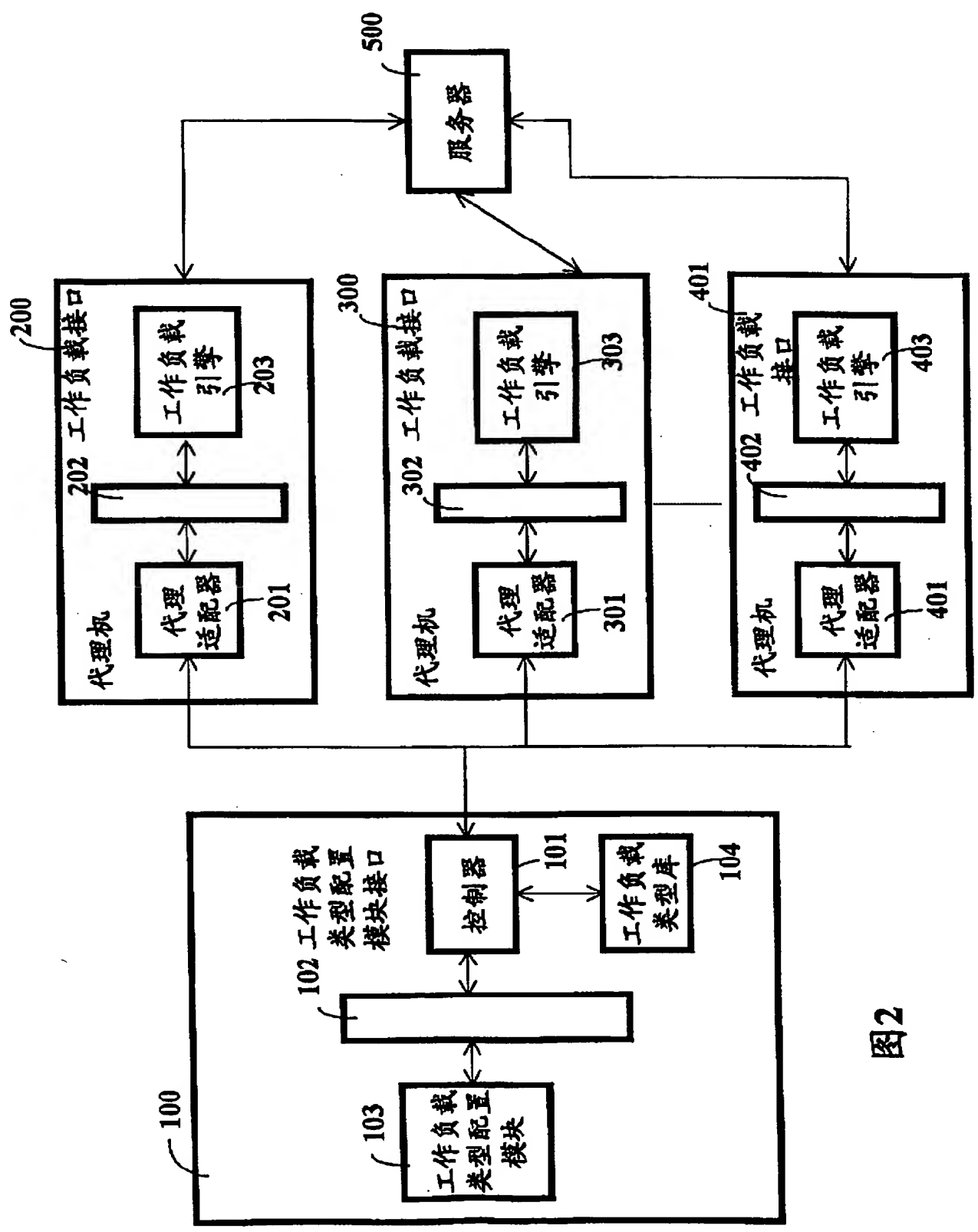


图2

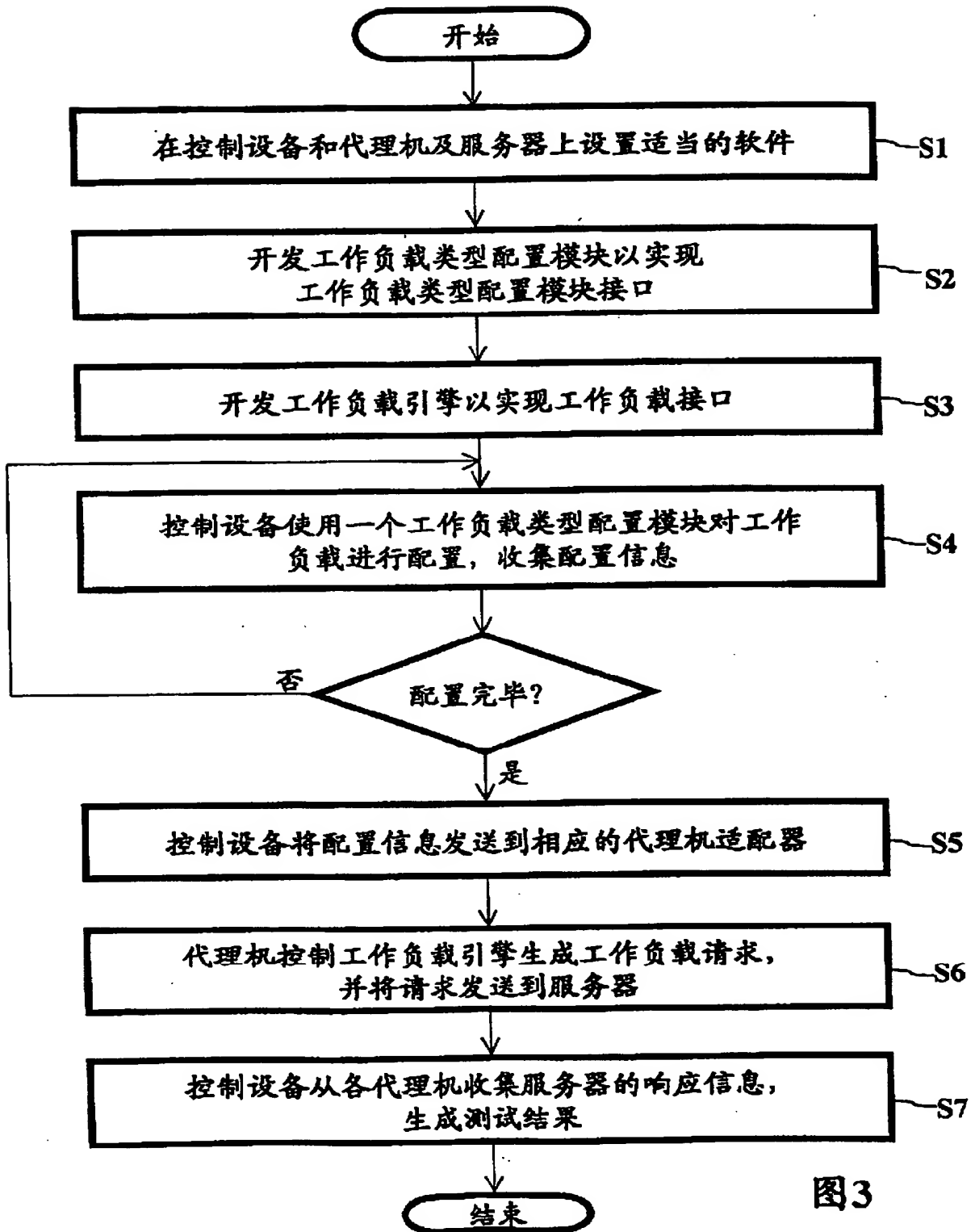


图3

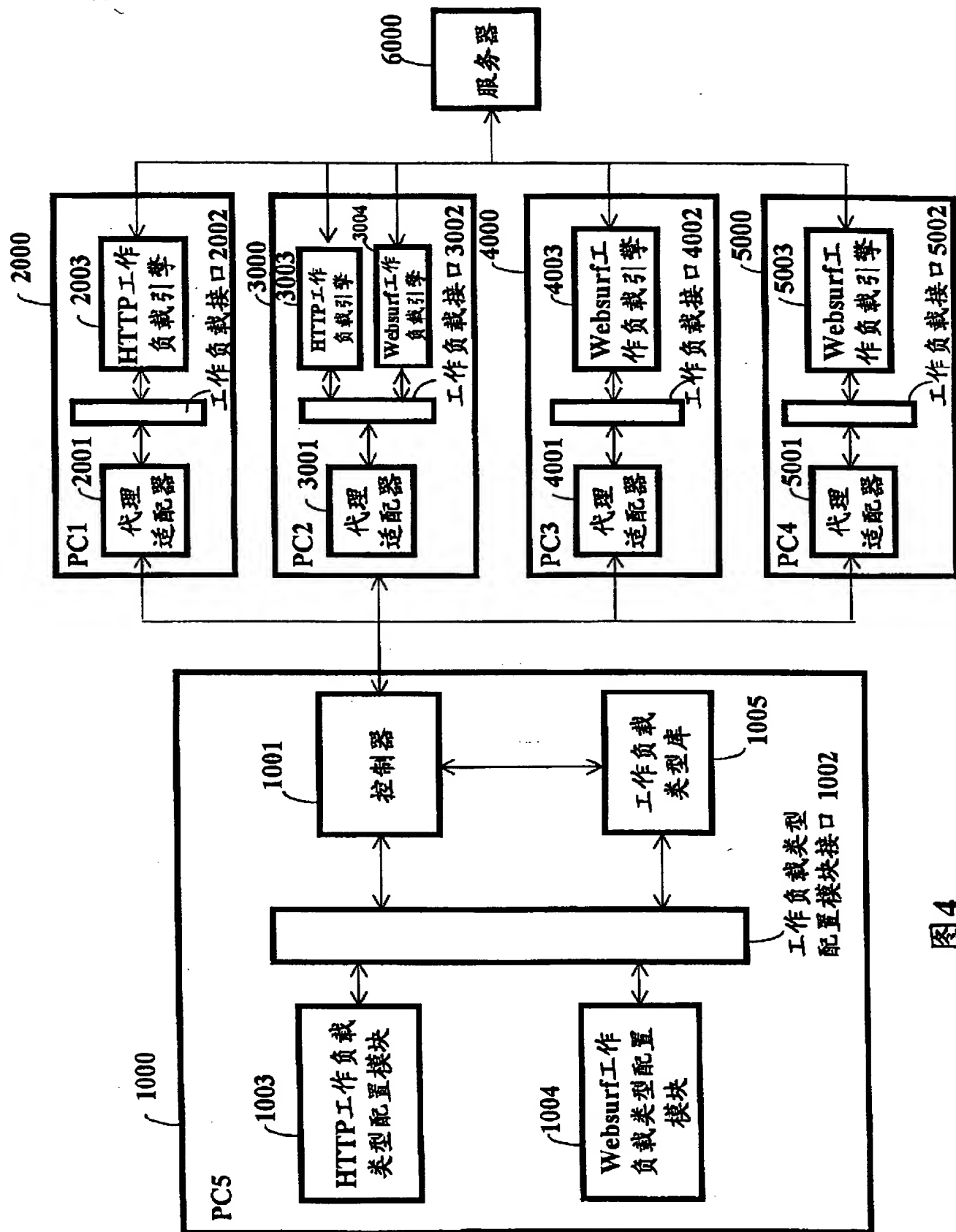


图4